UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/605,465	10/01/2003	Ivan Iliev Klianev		2464

38345 7590 12/14/2006

EXTRAPOLES PTY LIMITED
PO BOX 324
BRIGHTON-LE-SANDS NSW, 2216
AUSTRALIA

EXAMINER	
YAARY, MICHAEL D	

ART UNIT	PAPER NUMBER
2196	

DATE MAILED: 12/14/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/605,465	KLIANEV, IVAN ILIEV	
	Examiner	Art Unit	
	Michael Yaary	2196	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☒ Claim(s) *2* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10/01/2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Detailed Action

1. Claims 1-20 are pending in the application.

Claim Objections

Claim 1 recites the limitation "said running program code" in line 9. There is insufficient antecedent basis for this limitation in the claim.

Claim 2 recites the limitation “said workflow-activities” in lines 5-6. There is insufficient antecedent basis for this limitation in the claim.

Claim 2 recites the limitation “all participating elements of work” in lines 7-8. There is insufficient antecedent basis for this limitation in the claim.

Claim 2 recites the limitation “said item” in line 8. There is insufficient antecedent basis for this limitation in the claim.

Claim 2 recites the limitation “a said workflow-activity” in lines 9-10. There is insufficient antecedent basis for this limitation in the claim.

Claim 2 recites the limitation “said step” in line 22. There is insufficient antecedent basis for this limitation in the claim.

Claim 2 recites the limitation “said node” in line 52. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-17 are rejected under U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Each claim is directed to "A computer based method and article of manufacture," which is not covered under a process, machine, manufacture, or composition of matter, rendering the claims non-statutory.

Claims 18-20 are rejected under U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Each claim is directed to "A computer program product," which appears to represent functional descriptive material, software per se, and is therefore ineligible for protection.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claim 1-17 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1-17 are directed toward "A computer based method and article of manufacture." This usage of both method and article of manufacture provides a vague and an unclear description of the claimed subject matter.

Claims 1-17 are directed to "A computer based method and article of manufacture." The claims do not include method steps for a claimed method, nor members of structure for an article of manufacture.

In Claim 2, part B (iii), part (C), and part C (iii) use the word "optionally" when disclosing the limitation. This provides a vague, indefinite, and an uncertainty as to whether or not the limitations are necessarily required.

In claim 5, part B the word optionally is used when disclosing the limitation. This provides a vague, indefinite, and an uncertainty as to whether or not the limitations are necessarily required.

In claim 2, part A (ii) and (iii), and part B (i) the limitations recited are the "preferred embodiment." Without clearly identifying the preferred embodiment, this provides a vague, indefinite, and an uncertainty as to what is the preferred embodiment.

In claim 2, part A (ii) the limitation recited is "this workflow-process." This provides a vague, indefinite, and an uncertainty as to which workflow-process the limitation is referring to.

In claim 2, part A (ii) the limitation recited is "might be executed." This provides a vague, indefinite, and an uncertainty as to whether or not the limitation is necessarily required.

In claim 2, part C the limitation recited is "might initiate." This provides a vague, indefinite, and an uncertainty as to whether or not the limitation is necessarily required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 4 and 19 are rejected under 35 U.S.C. 102(b) as being anticipated by Cloud et al. (hereafter Cloud)(US Pat. 6,253,369).

Regarding claim 4, Cloud discloses a computer-based method and article of manufacture producing configuration of class objects and threads with capacity for concurrent processing of multitude of transactional workflow requests of identical type (column 10, lines 5-12 and lines 48-49 disclose multiple workflow objects dispatched to execute concurrently.), wherein said configuration of class objects and threads represents a workflow processing configuration corresponding to a particular workflow process description (column 10, lines 56-58), wherein said configuration of class objects and threads represents control flows between workflow-activities, including optional launch of parallel control flows within same workflow request processing and parallel control flows synchronization (column 9, lines 39-44 and column 10, lines 5-12), wherein said capacity for concurrent processing includes capacity of every one of

workflow activities participating in said workflow process for concurrent processing of multitude of transactional workflow requests (column 10, lines 7-9), wherein said configuration of class objects and threads has ability for adaptation to changing working conditions related to delays in distributed environment and fluctuations in received workload (figure 10, item 1060 and column 16, lines 11-18), wherein said adaptation has form of run-time self-adjustment of objects and threads configuration (column 16, lines 38-44).

Regarding claim 19, Cloud discloses a computer program product that executes as operating system service on a computer running under control of operating system capable of executing multiple execution threads (This is inherent in the abstract, lines 1-12, as any computer program product running on a computer will be running under a processor and under control of an operating system, and thus will also have capabilities of executing multiple execution threads), uses a facility for message queuing and transactional messaging for signaling requests for execution of workflow-activities (inherent in column 4, lines 2-5), uses a facility for distributed transaction coordination(inherent in column 3, line 64 – column 4, line 5), and enacts behavior and capabilities as per claim 4(column 3, line 64 – column 4, line 5).

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent

granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1 and 18 are rejected under 35 U.S.C. 102(e) as being anticipated by the applicant admitted prior art (Hereafter AAPA).

Regarding claim 1, AAPA discloses a computer-based method and article of manufacture for graphical development of fully executable workflow applications ([0007], lines 6-14); wherein said fully executable workflow applications are in form of computer-executable program code for loading in computer memory code segments ([0010], lines 1-8 and [0011], lines 4-6) and, after said workflow applications being run by invoking said program code loaded in code segments ([0007, lines 14-17)], said running program code facilitates construction of necessary objects and threads in a way reflecting desired workflow configuration interactively described during applications' graphical development ([0007, lines 14-17 and [0011], lines 6-10), wherein said necessary objects and threads facilitate processing of workflow orders in a way described during applications' graphical development ([0011], lines 10-16 disclose sufficient threads used during processing to avoid software related workflow bottlenecks.).

Regarding claim 18, AAPA discloses a computer program product for graphical development of fully executable workflow applications ([0007], lines 6-14).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claim 2 and 3 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA in view of Kim et al. (hereafter Kim)(US Pub. 2002/0065701).

Regarding claim 2, AAPA does not disclose defining of a workflow-process comprising the following steps:

(A) Defining workflow-activities matrix, wherein said workflow-activities are individual items of work comprised by potential workflow-process where all participating elements of work included in said item are potentially executed as a single transaction within a said workflow-activity of said potential workflow-process,

(B) Defining main flow-graph, wherein defined at step (A) workflow-activities further being represented as nodes of said flow-graph by incorporating control-connectors in order to represent a potential flow-of-control between said nodes within potential workflow-process, wherein each said control-connector potentially signals successful execution of its sourcing workflow-activity with execution result having value Y, wherein two or more control-connectors sourcing from one said node might initiate concurrent execution of destination nodes of said control-connectors, wherein two or more control-connectors having a common destination node impose a requirement for

synchronizing conjunction of signals of all control-connectors incoming to said common destination node in order to trigger execution of said common destination node,

(C) Optionally defining one or more alternative control-connectors, wherein each said alternative control-connector potentially signals non-successful execution of its sourcing workflow-activity with execution result having value N being different from non-successful execution result having value T requiring termination of processing of a particular workflow request, wherein two or more alternative control-connectors sourcing from one workflow-activity might initiate concurrent execution of destination workflow-activities of said alternative control-connectors, wherein a workflow-activity having established at step (B) a requirement for synchronizing conjunction of signals of all incoming to it control-connectors cannot be a destination workflow-activity of an alternative control-connector,

(D) Defining workflow-components matrix, wherein every element of said workflow-components matrix represents a software component, associated with a workflow-activity of workflow-process being defined, for potential plugging for execution as part of a single transaction within said workflow-activity of potential said workflow-process.

However, Kim discloses a workflow-process comprising following steps:

(A) Defining workflow-activities matrix (this is an inherently needed part of defining a workflow process as a matrix of workflow activities is needed to construct a workflow process and to facilitate the activities of the workflow process.), wherein said workflow-activities are individual items of comprised by potential workflow-process work

([0071], lines 3-5 and [0072], lines 2-3. A workflow matrix is inherent in defining a workflow process.), where all participating elements of work included in said item are potentially executed as a single transaction within a said workflow-activity of said potential workflow-process ([0072], lines 3-8),

(B) Defining main flow-graph (Flow-graph in figure 19), wherein defined at step (A) workflow-activities further being represented as nodes of said flow-graph by incorporating control-connectors in order to represent a potential flow-of-control between said nodes within potential workflow-process ([0072], lines 2-4 disclose a node representing an activity and an arrow (control-connector) representing a transition), wherein each said control-connector potentially signals successful execution of its sourcing workflow-activity with execution result having value Y, wherein two or more control-connectors sourcing from one said node might initiate concurrent execution of destination nodes of said control-connectors, wherein two or more control-connectors having a common destination node impose a requirement for synchronizing conjunction of signals of all control-connectors incoming to said common destination node in order to trigger execution of said common destination node (These are all inherent result analysis of workflow activities from the flow graph, based on the management and flow control of the workflow activities.):

(C) Optionally defining one or more alternative control-connectors, wherein each said alternative control-connector potentially signals non-successful execution of its sourcing workflow-activity with execution result having value N being different from non-successful execution result having value T requiring termination of processing of a

particular workflow request (transition 6 of figure 20, discloses an alternate transition route, alternative control-connector), wherein two or more alternative control-connectors sourcing from one workflow-activity might initiate concurrent execution of destination workflow-activities of said alternative control-connectors, wherein a workflow-activity having established at step (B) a requirement for synchronizing conjunction of signals of all incoming to it control-connectors cannot be a destination workflow-activity of an alternative control-connector (These are inherent results of the flow control of two or more alternative control-connectors sourcing from one workflow-activity.)

(D) Defining workflow-components matrix, wherein every element of said workflow-components matrix represents a software component, associated with a workflow-activity of workflow-process being defined, for potential plugging for execution as part of a single transaction within said workflow-activity of potential said workflow-process ([0070], lines 1-4)

It is also inherent in Kim's disclosure that defining constants, declaring and initializing two-dimensional arrays are integral parts of a program for an automated workflow process method as disclosed by Kim.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the AAPA, by implementing a method of defining a workflow process as taught by Kim, for the benefit of creating an efficient means of controlling and managing workflow processes.

Regarding claim 3, AAPA further discloses generation of source code and compiling and linking instructions [0009], lines 1-4), wherein said source code and

compiling and linking instructions are necessary and sufficient to build fully executable workflow application with third party produced and available on the market compilers and linkers ([0009, lines 1-4), after said source code is compiled and linked, and after produced executable program code is loaded and executed, said executable program code facilitates construction of hierarchical tree of objects reflecting specified, as per claim 2, workflow process description ([0007], lines 11-14).

10. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Cloud as applied to claim 4 above, and further in view of Du et al. (hereafter Du)(US Pat. 6,308,163).

Regarding claim 5, Cloud does not disclose the construction of a hierarchical tree of class objects with capacity to represent variety of workflow configurations, wherein said class objects are computer memory instances of classes or structures, wherein any level of said hierarchical tree contains one or multiple collections, each said collection containing predefined or dynamically defined number of one or multiple objects, wherein objects belonging to any said collection of same hierarchy level are computer memory instances of classes or structures of identical type, wherein first level of said hierarchical tree consists of a single collection and every next level of hierarchy contains one or multiple collections and possibly individual objects, wherein said collections and said individual objects being accessible via references controlled by

objects belonging to collections from previous level, wherein said hierarchical tree of class objects contains various levels and siblings:

However, Du discloses construction of a hierarchical tree of class objects with capacity to represent variety of workflow configurations (column 6, lines 62-65), wherein said class objects are computer memory instances of classes or structures (This is inherent as class objects correspond directly to a block of computer memory.), wherein any level of said hierarchical tree contains one or multiple collections (column 8, lines 3-7), each said collection containing predefined or dynamically defined number of one or multiple objects (column 8, lines 5-7), wherein objects belonging to any said collection of same hierarchy level are computer memory instances of classes or structures of identical type (This is inherent as class objects correspond directly to a block of computer memory.), wherein first level of said hierarchical tree consists of a single collection (column 7, lines 8-11) and every next level of hierarchy contains one or multiple collections and possibly individual objects (system 36 of figure 2 and column 7, lines 12-13), wherein said collections and said individual objects being accessible via references controlled by objects belonging to collections from previous level (column 7, line 21 and line 42 disclose levels interacting with previous levels), wherein said hierarchical tree of class objects contains following levels and siblings:

(A) First level, a steps-collection, wherein said collection objects represent steps of said workflow-process, each said object being parent of a Second level collection (column 7, lines 8-11 disclose a first level providing an overview of the processing capabilities of the workflow resource system);

(B) Second level, an activities-collection per step of said workflow-process, wherein said collection objects represent workflow-activities that might be executed concurrently with all other workflow-activities represented by objects belonging to same collection (column 7, lines 12-14), wherein each said object is parent of Third level siblings, said siblings being of 3 distinct categories: a processor collection, a workload-balancer object, and optionally a workflow-synchronization object;

(C) Third level's sibling One, a processors-collection per workflow-activity, wherein an object of said collection represents an individual workload-processing squad having a capacity to process a portion of entire workload specific for said workflow-activity, each said object being parent of a Fourth level collection;

(D) Third level's sibling Two, a workload-dispatching object per workflow-activity;

(E) Third level's optional sibling Three, a workflow-synchronizing object per workflow-activity if required by flow-graph of said workflow-process, wherein said object contains data defining required synchronization scheme (Column 7, lines 25-41 disclose a third level of multiple local resource managers (LRMs, which would be sibilings) controlling and managing the different resources, or workflow processes. This therefore, just an obvious variation of the third-level siblings of the instant application.);

(F) Fourth level, a processing-thread-dedicated objects collection, where an object of said collection represents a dedicated area for data holding and data exchange between threads (Figure 3, fourth level 94).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention, to modify Cloud by constructing a hierarchical tree with capacity to

represent a variety of workflow configurations as taught by Du, for the benefit of having an efficient management system facilitating the flow of workflow resources.

10. Claims 6-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cloud and Du as applied to claim 5 above, and further in view of Hsu et al. (hereafter Hsu)(US Pat. 5,581,691).

Regarding claim 6, Cloud and Du do not disclose a method of splitting of a control flow by launching one or more new parallel control flows, wherein said parallel control flow comprises execution of at least one workflow-activity, or sequence of plurality of workflow-activities, before joining with its launching, or other, flow of control, wherein said parallel control flow might optionally launch one or more new parallel control flows, wherein a workflow-activity, that is part of a parallel control flow, generates notification message to synchronizing thread at control flow join-point with other control flow or plurality of control flows about a potential result with value T of its potential execution, wherein a workflow-activity, that is part of a parallel control flow further splitting one or more times, generates notification messages to synchronizing threads at splitting branches join-points with other control flows about a potential result with value T of its potential execution, wherein synchronizing threads generate notification messages to synchronizing thread responsible for join-point of synchronized control flow with one or more control flows or to synchronizing threads responsible for join-points of said synchronized control flow's splitting branches about potential termination of said synchronized control flow.

However, Hsu discloses a method of splitting of a control flow by launching one or more new parallel control flows (flow x of figure 3, and column 5, lines 25-27), wherein said parallel control flow comprises execution of at least one workflow-activity, or sequence of plurality of workflow-activities, before joining with its launching, or other, flow of control, wherein said parallel control flow might optionally launch one or more new parallel control flows (flow x of figure 3, and column 5,lines 51-52), wherein a workflow-activity, that is part of a parallel control flow, generates notification message to synchronizing thread at control flow join-point with other control flow or plurality of control flows about a potential result with value T of its potential execution, wherein a workflow-activity, that is part of a parallel control flow further splitting one or more times, generates notification messages to synchronizing threads at splitting branches join-points with other control flows about a potential result with value T of its potential execution, wherein synchronizing threads generate notification messages to synchronizing thread responsible for join-point of synchronized control flow with one or more control flows or to synchronizing threads responsible for join-points of said synchronized control flow's splitting branches about potential termination of said synchronized control flow (column 5, lines 12-13 disclose a flow controller that is responsible for the transmission of messages. Thus the notification messages generated by split workflow activities are none other than inherent communication procedures that must be done when splitting and processing parallel workflow activities.)

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cloud and Du, by including a method of splitting of a control flow by launching one or more new parallel control flows as taught by Hsu, for the benefit of reducing the amount of time required to complete a work flow (Hsu column 5, lines 57-58).

Regarding claim 7, Cloud and Du do not disclose a method for synchronization of two or more parallel control flows before execution of next in flow-graph workflow-activity according to a synchronization scheme, wherein said synchronization scheme is a conjunction of events signaling completed execution of all workflow-activities represented by sourcing nodes of two or more flow-graph control connectors with destination node being said workflow-activity that will be executed after said synchronization as part of a potential single workflow-request, wherein implementation of said synchronization method applies to entire multitude of concurrently processed potential workflow-requests.

However, Hsu discloses a method for synchronization of two or more parallel control flows before execution of next in flow-graph workflow-activity according to a synchronization scheme, wherein said synchronization scheme is a conjunction of events signaling completed execution of all workflow-activities represented by sourcing nodes of two or more flow-graph control connectors with destination node being said workflow-activity that will be executed after said synchronization as part of a potential single workflow-request, wherein implementation of said synchronization method

applies to entire multitude of concurrently processed potential workflow-requests (figure 3, steps 152-1,152-2,152-3 and column 5, lines 52-56 disclose parallel paths (control flows) being performed simultaneously, therefore, being synchronized prior to the execution of the next workflow activity.)

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cloud and Du, by including a method for synchronization of two or more parallel control flows before execution of next in flow-graph workflow-activity as taught by Hsu, for the benefit of reducing the amount of time required to complete a work flow (Hsu column 5, lines 57-58).

Regarding claim 8, Cloud and Du do not disclose a method of launching alternative control flow routes, wherein said launching optionally takes place where potential execution of a workflow-activity has a result with value N, wherein said launching redirects workflow execution from its normal routes by transferring execution control to one or more workflow-activities with non-synchronized execution, wherein an alternative control-connector launching an alternative control flow route and bypassing one or more workflow-activities with synchronized execution, is coupled with notification-connectors having same source node as said alternative control-connector, wherein each one of said notification-connectors has one of bypassed workflow-activities with synchronized execution as its destination node.

However Hsu discloses a method of launching alternative control flow routes, wherein said launching optionally takes place where potential execution of a workflow-activity has a result with value N, wherein said launching redirects workflow execution

from its normal routes by transferring execution control to one or more workflow-activities with non-synchronized execution, wherein an alternative control-connector launching an alternative control flow route and bypassing one or more workflow-activities with synchronized execution, is coupled with notification-connectors having same source node as said alternative control-connector, wherein each one of said notification-connectors has one of bypassed workflow-activities with synchronized execution as its destination node (column 11, lines 1-4 and lines 10-29 disclose the process of implementing an alternative route based on the output of two different quality values, thus reading on launching an alternative control flow route where potential execution of a workflow-activity has a result with value N . This is also shown in figure 12 where the option is to take the route leading to step 474 or step 476.

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cloud and Du, by including a method of launching alternative control flow routes as taught by Hsu, in order to further facilitate and allow for a well-managed flow of workflow activities.

Regarding claim 9, Cloud and Du do not disclose construction of hierarchical structure of threads with four levels, wherein in said hierarchical structure levels below top level are organized as multitude of horizontally arranged divisions, wherein each said division is autonomous and self-contained in conducting its tasks, wherein top level thread is responsible for making adaptive decisions, and executing and supervising adaptive behavior related to allocation and de-allocation of system resources based on its own assessment of application needs and goals, wherein said structure of threads

provides capacity for concurrent processing of multitude of requests, limited only by environmental factors such as availability of reserve of system memory and unused CPU power and ability of networking infrastructure to cope with generated traffic, wherein said hierarchical structure of threads contains several levels.

However, Hsu discloses construction of hierarchical structure of threads with four levels, wherein in said hierarchical structure levels below top level are organized as multitude of horizontally arranged divisions, wherein each said division is autonomous and self-contained in conducting its tasks, wherein top level thread is responsible for making adaptive decisions, and executing and supervising adaptive behavior related to allocation and de-allocation of system resources based on its own assessment of application needs and goals, wherein said structure of threads provides capacity for concurrent processing of multitude of requests, limited only by environmental factors such as availability of reserve of system memory and unused CPU power and ability of networking infrastructure to cope with generated traffic, wherein said hierarchical structure of threads contains following levels (four layer architecture of figure 4):

(A) First level, formed by created and activated processing threads, wherein threads forming this level of said hierarchical structure are directly responsible for transactional processing of requests for work and for transactional flow of control between workflow-activities (interface layer of figure 4 and column 8, lines 41-52);

(B) Second level, formed by created and activated supervising threads, wherein threads forming this level of said hierarchical structure are responsible

for assignment of requests to individual processing threads and supervision of requests' execution (policy manager and resource model layer of figure 4 and column 8, lines 53-60);

(C) Third level, formed by all dispatching or synchronizing-dispatching threads being created and activated according to number of workload-dispatching objects in hierarchical tree of class objects and number of optional workflow-synchronizing objects in hierarchical tree of class objects, wherein dispatching threads being part of Third level are responsible for dispatching of requests to supervising threads, wherein synchronizing-dispatching threads being part of Third level are responsible for synchronization of execution of parallel control flows and for dispatching of requests to supervising threads (request processing engine layer of figure 4, and column 8, lines 61-65);

(D) Fourth level, wherein Fourth level is hierarchy's top level and comprises only one thread being responsible for making, executing and supervising decisions about allocation and de-allocation of system resources based on its own assessment, wherein said allocation and de-allocation takes form of modification of First and Second levels of hierarchical structure of threads and their corresponding objects and collections of hierarchical tree of class objects (integration layer of figure 4 and column 8, lines 67-column 9, lines 1-5).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cloud and Du, by including construction of

hierarchical structure of threads with four levels as taught by Hsu, in order to further facilitate and allow for a well-managed flow of workflow activities.

11. Claims 10-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cloud, Du, and Hsu as applied to claim 9 above, and further in view of LiVecchi (US Pat. 6,823,515).

Regarding claim 10, Cloud, Du, and Hsu do not disclose a method for transactional plugging of software components into workflow-process comprising following steps:

(A) Creation and initialization of processing threads, wherein during its initialization each said processing thread instantiates a non-transactional component object and sends to it an amount of data being constant between processing of individual workflow requests and being necessary to perform execution of relevant portion of workflow request that will potentially be assigned to said processing thread;

(B) Instantiated at step (A) non-transactional component object instantiates its own transactional component-intercepting object, and with GUID (Globally Unique Identifier), sent to it as parameter, instantiates a workflow-activity component-intercepting object;

(C) Supervising thread receives potentially arriving multitudes of single workflow requests and assigns each one of said requests to individual processing thread of its pool;

(D) A processing thread having an assigned at step (C) request for work invokes a method of its non-transactional component object;

(E) An invoked at step (D) non-transactional component object method calls a method of its transactional component-intercepting object, wherein said call of transactional component-intercepting object method constructs transactional component object thereby creating a new transaction and calls a method of said transactional component;

(F) Within context of created at step (E) transaction, said transactional component object method calls a method of instantiated at step (B) workflow-activity component-intercepting object, wherein said method call instantiates a workflow-activity software component and invokes a method of said component.

However, LiVecchi discloses a method for transactional plugging of software components into workflow-process (abstract) comprising following steps:

(A) Creation and initialization of processing threads, wherein during its initialization each said processing thread instantiates a non-transactional component object and sends to it an amount of data being constant between processing of individual workflow requests and being necessary to perform execution of relevant portion of workflow request that will potentially be assigned to said processing thread (Column 2, lines 46-54 disclose how a pool of processing threads is created, thus initialized, in order to deal with the load of workflow requests.);

(B) Instantiated at step (A) non-transactional component object instantiates its own transactional component-intercepting object, and with GUID (Globally Unique

Identifier), sent to it as parameter, instantiates a workflow-activity component-intercepting object (It would be obvious that a GUID would be used to identify and instantiate a workflow-activity component-intercepting object as GUIDs are commonly used in software applications);

(C) Supervising thread receives potentially arriving multitudes of single workflow requests and assigns each one of said requests to individual processing thread of its pool (column 5, lines 29-32);

(D) A processing thread having an assigned at step (C) request for work invokes a method of its non-transactional component object (column 7, lines 27-31 discloses the method of which a worker, processing, thread is implemented in the workflow processing);

(E) An invoked at step (D) non-transactional component object method calls a method of its transactional component-intercepting object, wherein said call of transactional component-intercepting object method constructs transactional component object thereby creating a new transaction and calls a method of said transactional component (It is inherent that the transactional component is called when implementing the workflow method, as it necessary for the processing of workflow activities.);

(F) Within context of created at step (E) transaction, said transactional component object method calls a method of instantiated at step (B) workflow-activity component-intercepting object, wherein said method call instantiates a workflow-activity software component and invokes a method of said component (It is inherent that the

method workflow activity is called when implementing the workflow method, as it necessary for the processing of workflow activities.).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cloud, Du, and Hsu by, including a method for transactional plugging of software components into workflow process, as taught by LiVecchi for the benefit of creating and managing an efficient workflow process.

Regarding claim 11, Cloud, Du, and Hsu do not disclose workload balancing structured at two levels, wherein upper level of said workload balancing comprises multitude of associations between a dispatching thread and multitude of supervising threads and involves dispatching thread balancing workload between its associated supervising threads, wherein lower level of said workload balancing comprises multiple groupings of processing threads in pools associated with a supervising thread per pool and involves supervising threads balancing workload between processing threads of their associated pools.

However, LiVecchi discloses workload balancing structured at two levels, wherein upper level of said workload balancing comprises multitude of associations between a dispatching thread and multitude of supervising threads and involves dispatching thread balancing workload between its associated supervising threads, wherein lower level of said workload balancing comprises multiple groupings of processing threads in pools associated with a supervising thread per pool and involves supervising threads balancing workload between processing threads of their associated pools (column 3, lines 24-32 and column 5, lines 27-35 disclose the method in which

dispatching threads are used to assign workflow requests to worker, or processing, threads in order to balance the workload.)

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cloud, Du, and Hsu, by including a workload balancing as taught by LiVecchi, for the benefit of enhancing workload performance being done, and avoiding delays in the workload flow.

Regarding claim 12, Cloud, Du, and Hsu do not disclose software bottlenecks' prevention and neutralizing, wherein said software bottlenecks' prevention involves encapsulation of a thread pool containing fixed number of processing threads with a supervising thread in a processing-pipe, wherein said software bottlenecks' neutralizing comprises construction of additional processing-pipes and inclusion of constructed additional processing-pipes in workload balancing process related to workflow-activity where development of bottleneck has been detected.

However, LiVecchi discloses software bottlenecks' prevention and neutralizing, wherein said software bottlenecks' prevention involves encapsulation of a thread pool containing fixed number of processing threads with a supervising thread in a processing-pipe (column 2, lines 46-48), wherein said software bottlenecks' neutralizing comprises construction of additional processing-pipes and inclusion of constructed additional processing-pipes in workload balancing process related to workflow-activity where development of bottleneck has been detected (column 16, lines 43-45 and lines 56-59 disclose the creation of a collector socket to help facilitate workflow. This is a beneficial process, as the problem in the prior art is relates to over-scheduling of

threads to service workloads, thus creating inefficient use of resources. Thus, being analogous to neutralizing and preventing a software bottleneck.)

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cloud, Du, and Hsu, by including software bottlenecks' prevention and neutralizing as taught by LiVecchi, for the benefit of enhancing the performance of running multi-threaded applications.

Regarding claim 13, Cloud, Du, and Hsu do not disclose a method and apparatus for automatic detection of conditions requiring workflow application scaling up, wherein said automatic detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of idle threads in processing-pipe's pool reached its critical minimum.

However, LiVecchi discloses a method and apparatus for automatic detection of conditions requiring workflow application scaling up, wherein said automatic detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of idle threads in processing-pipe's pool reached its critical minimum (column 12, lines 66-67 – column 13, lines 1-28 disclose a scheduling heuristic used to determine whether more threads should be used to process a workload connection, thus being analogous to determining whether scaling up of workflow applications is necessary.

Therefore, it would have been obvious to one of ordinary skill at the time of the invention to modify, Cloud, Du, and Hsu, by including method and apparatus for

automatic detection of conditions requiring workflow application scaling up as taught by LiVecchi, for the benefit of enhancing the performance of running multi-threaded applications.

Regarding claim 14, Cloud, Du, and Hsu do not disclose a method and apparatus for automatic workflow application scaling up, wherein said application scaling up is automatically triggered at a particular workflow-activity to counteract development of a bottleneck at that particular workflow-activity and automatically triggered at all application's workflow-activities for higher application responsiveness when workload increases, wherein said application scaling up involves creation and activation of an additional processing-pipe and inclusion of said additional processing-pipe in workload balancing scheme.

However, LiVecchi discloses including a and apparatus for automatic workflow application scaling up, wherein said application scaling up is automatically triggered at a particular workflow-activity to counteract development of a bottleneck at that particular workflow-activity and automatically triggered at all application's workflow-activities for higher application responsiveness when workload increases, wherein said application scaling up involves creation and activation of an additional processing-pipe and inclusion of said additional processing-pipe in workload balancing scheme (column 16, lines 43-50 disclose the creation of a collector socket to be created when necessary for thread processing, thus alleviating the problem of over scheduling of threads and as a result not causing delays and software bottlenecks).

Therefore, it would have been obvious to one of ordinary skill at the time of the invention to modify Cloud, Du, and Hsu by, including method and apparatus for automatic workflow scaling up as taught by LiVecchi, for the benefit of enhancing the performance of running multi-threaded applications.

Regarding claim 15, Cloud, Du, and Hsu do not disclose a method and apparatus for automatic detection of conditions requiring workflow application scaling down, wherein said automatic detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of busy threads in processing-pipe's pool reached its critical minimum.

However, LiVecchi discloses a method and apparatus for automatic detection of conditions requiring workflow application scaling down, wherein said automatic detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of busy threads in processing-pipe's pool reached its critical minimum (column 19, lines 15-28 disclose detecting whether an idle control socket connection should be removed based on when the idle connection reaches and passes a maximum value. Therefore, the number of busy threads would be at a minimum, and as a result reads on detecting conditions requiring scaling down of workflow applications.)

Therefore, it would have been obvious to one of ordinary skill at the time of the invention to modify Cloud, Du, and Hsu by, employing detection of conditions requiring

workflow application scaling down as taught by LiVecchi, for the benefit of freeing up unnecessarily used resources.

Regarding claim 16, Cloud, Du, and Hsu do not disclose including method and apparatus for automatic workflow application scaling down, wherein said application scaling down is automatically triggered to counteract a detected inefficiency in use of system memory and CPU time slice allocated to application threads.

However, LiVecchi discloses a method and apparatus for automatic workflow application scaling down, wherein said application scaling down is automatically triggered to counteract a detected inefficiency in use of system memory and CPU time slice allocated to application threads (Column 19, lines 25-26 and column 20, lines 31-33 disclose closing down the connection, or scaling down, in order to free up system resources, thus including the system's memory and CPU.).

Therefore, it would have been obvious to one of ordinary skill at the time of the invention to modify Cloud, Du, and Hsu by including method and apparatus for automatic workflow application scaling down as taught by LiVecchi, in order to allow system resources to run more efficiently.

12. Claims 17 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cloud, Du, Hsu, and LiVecchi as applied to claim 16 above, and further in view of Lewis et al. (herafter Lewis)(US Pat 6,434,714).

Regarding claims 17 and 20, Cloud, Du, Hsu, and LiVecchi do not disclose real-time visualization of quantity, structure, and utilization of threads forming First and

Second levels of hierarchical structure of threads and said hierarchical structure's adaptation-enacted modifications of its First and Second levels, wherein said visualization might be used as indicator of workload, indicator of points of delay caused by distributed infrastructure, and for observation and analysis of adaptive behavior of hierarchical structure of threads.

However, Lewis discloses real-time visualization of quantity, structure, and utilization of threads forming First and Second levels of hierarchical structure of threads and said hierarchical structure's adaptation-enacted modifications of its First and Second levels, wherein said visualization might be used as indicator of workload, indicator of points of delay caused by distributed infrastructure, and for observation and analysis of adaptive behavior of hierarchical structure of threads (display 400 of Figure 4 and column 5, lines 51-58 disclose a graphical analysis of workflow application performance based on time, thus inclusive of quantity, structure, performance and utilization.)

Therefore, it would have been obvious to one of ordinary skill at the time of the invention to modify Cloud, Du, Hsu, and LiVecchi, by including real-time visualization of quantity, structure, and utilization of threads as taught by Lewis, for the benefit of determining the achievement of optimal thread performance (Lewis column 1, lines 48-50).

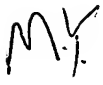
Conclusions

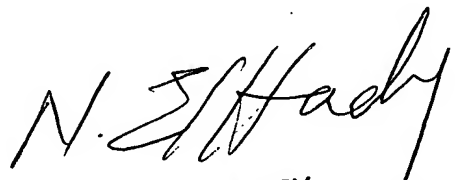
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Yaary whose telephone number is (571) 270-1249. The examiner can normally be reached on Monday-Friday, 8:00 a.m - 5:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nabil El-Hady can be reached on (571) 272-3963. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MY





NABIL M. EL-HADY
SUPERVISORY PATENT EXAMINER